Proceedings of the 2012 IEEE 12th International Conference on Bioinformatics & Bioengineering (BIBE), Larnaca, Cyprus, 11-13 November 2012

A New Iterative Graph Based Clustering Method

Erion-Vasilis Pikoulis and Emmanouil Z. Psarakis
Department of Computer Engineering and Informatics, University of Patras, 26500 Rio-Patras, Greece
email:{pikoulis, psarakis}@ceid.upatras.gr
phone: +30 2610 996969, fax: +30 2610 996971

***Abstract*—In this paper we propose a new iterative graph based clustering technique. The proposed method has three desirable characteristics as compared to well known clustering techniques. Specifically, because of its iterative nature, the number of clusters contained in a given data set it is not necessary to be known a priori as partitional clustering techniques demand. Its performance, in terms of the quality of the achieved clustering, does not depend on the distribution of the cluster's size. Finally, no threshold value is required for achieving the clustering as hierarchical clustering techniques demand. All these features manifest themselves in the important problem of clustering homologous proteins when only sequence information is available. The proposed method is tested against well known and widely used techniques, by conducting a number of experiments based on both artificial and real protein data sets and all above mentioned characteristics were confirmed.**

I. Introduction

An important problem in today's genomics is that of grouping together homologous proteins when only sequence information is available. This is a difficult problem since sequence similarity is a very noisy measure of evolutionary relatedness [1]. The core of most methods proposed so far in the bioinformatics literature, is based on well known clustering techniques also used in other fields of sciences.

The importance of data clustering, the great number and diversity of applications, along with the specific requirements each of them poses, combined with the lack of a universally accepted definition of the term cluster, has led to a plethora of clustering methods over the past decades. In recent years especially, the computer revolution has provided scientists with very large amounts of data and the computation resources to process and analyse them, leading to the development of modern clustering techniques. The existence of this great number of clustering methods, gives an indication to what nowadays is generally accepted, that there exists no globally better method. Instead, each method has its strengths and weaknesses and its best suited to a specific class of applications [2].

Although as mentioned above, the definition of a cluster is somewhat broad and takes usually the shape of the specific requirements of the application at hand, the general intuition behind it is that clusters are groups of objects presenting similar characteristics among them, and dissimilar to the rest of the data set. If a data set comprises of well-defined and well-separated groups, then the cluster identification problem becomes a fairly trivial one and the vast majority of the proposed solutions would yield very similar results. In real applications however, this is seldom the case, as real data sets are usually characterized by inhomogeneity in their inter-object relations, having strongly related groups co-existing or possibly overlapping with more loosely connected groups, as well as "irrelevant" objects. What makes the problem even more difficult is that, in the majority of real applications, this knowledge is not present a priori and safe assumptions can not be made. In these cases, the particular choice of the clustering method used is of crucial importance, and different methods applied to the same data set could yield highly different results. Hence, it becomes clear that the selected method should comply with the clustering requirements of the application it is intended to, i.e. the interpretation given by the method to the clustering objective, as well as the goals it sets in order to achieve it, should match the needs of the application at hand.

On the basis of the aforementioned interpretation of the problem, as well as the a priori knowledge required from them, the main bulk of the existing clustering methods can be loosely classified into two broad categories; namely the partitional and the hierarchical ones.

In partitional clustering methods, the data set at hand is partitioned into a predefined number of clusters, by seeking the partition that optimizes a clustering quality measure (e.g modularity [3]). The most popular representative of this class of methods is k-means [4], where the data are represented as points in Euclidean space and the goal of the method is to find the partition that minimizes the sum of distances of each object from the center mass of the cluster it is assigned to. In the same category belongs the family of spectral clustering methods, which represent the given objects and the relations between them in the form of a similarity graph, and seek to partition the graph nodes into k disjoint groups, by introducing suitable clustering objectives based on the spectral characteristics of the graph, such as ratio cut (RCut), normalized cut (NCut), and ratio association (RAssoc) [5], [6], [7]. It has been recently shown, that the above mentioned spectral clustering techniques are equivalent to the kernel k-means one [8]. In both of the above families of methods, optimization leads to NP-hard problems and solution is obtained through relaxation of the criteria, or heuristic algorithms. The main disadvantage of partitional clustering methods though, is that the number of clusters, k, required beforehand, is in most real cases unknown and difficult to assume, since in most applications, the only knowledge one is provided with, is the data set itself. Moreover, there is a resolution limit when clustering is based on global (i.e. affected by the whole data set) optimization criteria, meaning that small sized clusters are often not identified [2] [9].

The family of the hierarchical methods on the other hand, can be considered as the backbone of the clustering methods that are used today in various applications, with the class of the agglomerative methods being the most popular one [10]. Instead of producing a single partition of data into clusters, the output of this class of methods is a whole

978-1-4673-4358-9/12/$31.00 ©2012 IEEE

hierarchy of partitions, each one resulting by merging a pair of suitably selected clusters of the previous partition. Each level of the hierarchy is assigned a constant, equalling the similarity of the clusters being merged at that particular point. The outcome, called hierarchical clustering scheme, is usually represented graphically by a dendrogram that concentrates all the information of the various clustering stages, from the lowest (leaf) level, where every object belongs to a different cluster, up to the highest (root) level, where all the objects belong to the same cluster [11]. What differentiates the various representatives of this family of methods, is the way each one defines inter-cluster similarity. More specifically, single *linkage clustering* defines similarity between clusters $\mathcal{A}$, $\mathcal{B}$, as the similarity of the "closest" or most similar pair of objects (x, y), with $x \in \mathcal{A}$ and $y \in \mathcal{B}$. At the other end of the spectrum, *complete linkage* clustering defines similarity between $\mathcal{A}$, $\mathcal{B}$, by the "farthest" or most dissimilar pair (x, y), with $x \in \mathcal{A}$ and $y \in \mathcal{B}$. Finally, *average linkage* clustering is a compromise between these two extremes, defining inter-cluster similarity as the mean of the pairwise similarities of all (x, y), with $x \in \mathcal{A}$ and $y \in \mathcal{B}$. Although hierarchical clustering has the advantage that it does not require a priori knowledge on the number and size of the clusters, it presents itself with a number of significant disadvantages as well. The first and most obvious one is that the hierarchical structure produced by this family of methods could be a rather artificial representation of the data set at hand, if the latter does not possess this kind of structure in the first place, which is the case in most real applications. Moreover, there is no clear way to determine, of all the partitions it produces, the one that best represents the structure of the data set at hand. This is usually done by truncating the hierarchy to a selected level, using a threshold value, which in most cases is chosen empirically. Furthermore, regardless of the particular hierarchy level, there are some inherent issues, attributed to the merging criteria used, that often lead to well known and documented misclassification problems (such as the "chaining" effect for single linkage and "scattering" for complete linkage clustering) [10].

Finally, though a great number of techniques have been proposed regarding the particular problem of sequence-based protein clustering, (e.g TransClust [12] or HiFix [13]), the most prominent and one of the most used clustering algorithms in bioinformatics is the Markov Clustering Algorithm [14]. MCL simulates a flow on the given graph by calculating successive powers of the associated adjacency matrix, a procedure which is known as *expansion* and is responsible for allowing flow to connect different regions of the graph. In addition, at each iteration, an *inflation* step is applied, in order to enhance the existing contrast between regions of strong and weak flow in the graph. Assuming non-overlapping clusters with moderate diameters, the process converges towards a partition of the graph, with the set of high-flow regions (the clusters) separated by boundaries with no flow. Since the granularity of the resulting clustering is controlled by the inflation parameter, its value strongly affects the number and overall quality of the identified clusters [2] [14].

The paper is organised as follows. In Section II some graph theoretic preliminaries are presented. In Section III the proposed technique is analysed in detail. Section IV contains our experimental setup, the clustering results and the comparisons of the proposed technique with other well known methods. Finally, Section V contains our conclusions.

II. Preliminaries

Let us consider that the given data set is represented in terms of a similarity graph $\mathcal{G} = (\mathcal{V},\ \mathcal{E},\ W)$, where $\mathcal{V}$, $\mathcal{E}$ and W are the nodes, the edges sets and the weight matrix, respectively. In this clustering model, each node in $\mathcal{V}$ represents an object of the data set, and the edge weight between any two nodes represents the similarity of the corresponding objects. More specifically, the $W(i, j)$ element of the $|\mathcal{V}| \times |\mathcal{V}|$ matrix W contains the edge weight existing between nodes i and j and $|\mathcal{X}|$ denotes the cardinality of set $\mathcal{X}$. Let us now define the following quantity:

$$R(i, \mathcal{A}) = \frac{1}{|\mathcal{A}|} \sum_{j \in \mathcal{A}} W(i, j), \tag{1}$$

which quantifies the relevance of the $i-$th node, $i \in \mathcal{V}$, to a subset $\mathcal{A} \subseteq \mathcal{V}$. Using this quantity we can define the following quality measures for a cluster candidate subset of objects $\mathcal{V}_c \subseteq \mathcal{V}$:

$$Q(\mathcal{V}_c) = \min_{i \in \mathcal{V}_c} R(i, \mathcal{V}_c) \tag{2}$$

$$S(\mathcal{V}_c) = \max_{i \in \mathcal{V} \backslash \mathcal{V}_c} R(i, \mathcal{V}_c) \tag{3}$$

where $\mathcal{A} \backslash \mathcal{B}$ denotes the *relative complement* of set $\mathcal{B}$ in set $\mathcal{A}$ or equivalently the *set-theoretic difference* of $\mathcal{B}$ and $\mathcal{A}$.

Note that a high value of $Q(\mathcal{V}_c)$ ensures that every node of $\mathcal{V}_c$ is highly relevant to the other nodes of $\mathcal{V}_c$, and hence there is a high probability that the objects represented by $\mathcal{V}_c$ are part of the same class. On the other hand, if $\mathcal{V}_c$ contains even as much as one node with low relevance to the other members (a node that normally should be characterized as an outlier), then its presence will be reflected on the value of the measure.

Similarly, a low value of $S(\mathcal{V}_c)$ ensures that there exists no node outside $\mathcal{V}_c$ with high relevance to it, meaning that the subgraph $\mathcal{G}_c$ induced by $\mathcal{V}_c$ can be considered well-separated from the rest of the graph.

As it will become apparent in the next section, the above defined measures enable us to express the overall quality of a cluster in "worst case" terms, i.e. as a function of the most dissimilar included node and the most similar of the not included ones.

III. The proposed method

Let us now proceed to present the proposed method which, as we have already mentioned, is iterative with each iteration being a *two-step* procedure. Namely, the first step is an *elimination*, or *deconstruction*-step while the second is an *augmentation*-step.

A. *Elimination-Step*

Let us form the following sequence of node sets with a monotone decreasing cardinality:

$$\mathcal{V}_e(k) = \mathcal{V}_e(k-1) \backslash i_{min}(k), \quad \mathcal{V}_e(0) = \mathcal{V} \tag{4}$$

where

$$i_{min}(k) = \underset{i \in \mathcal{V}_e(k-1)}{\arg\min}\ R(i, \mathcal{V}_e(k-1)). \tag{5}$$

As it is clear from Equ. (5), the above sequence is obtained by repeatedly eliminating the node with the lowest relevance, of the ones currently present in set $\mathcal{V}_e(k-1)$ (i.e. the node that determines the relevance measure $Q(\mathcal{V}_e(k-1))$). Since the elimination of a node (as well as the edges incident to it) can only reduce the degrees of the nodes that were adjacent to it, it is clear that if the node eliminated on a given round is part of a dense and well separated group of nodes then, with high probability the majority of the reduced degrees will belong to nodes of the same group, which, due to the aforementioned assumption, had similar degrees with the eliminated node. As a result, the node that will be eliminated on the next round will most likely also belong to the same group, and this will carry on until the whole group is eliminated. Based on this, we expect the outcome of this step to be a rough grouping of the nodes, with the smallest groups located towards the beginning of the elimination sequence and the largest ones located towards the end of it. As we go deeper in the elimination rounds, the nodes that are still remaining towards the end of the procedure, should not only have higher degrees in the initial graph than the ones already eliminated, but they should also present high similarities among them, as it is precisely the large weights of the edges between them that allowed them to survive the elimination rounds. In this sense, the lastly eliminated nodes should not only belong to the same cluster, but they should also form the most compact (heavily connected) part of it, or its *core* (and the objects they represent form the core of the corresponding cluster). Based on this reasoning, the algorithm decides that has reached the core of a cluster, whenever quality measure $Q(\mathcal{V}_e(k))$ of the remaining subgraph exceeds a predetermined threshold $0 \leq \alpha \leq 1$. We must stress at this point that the selection of α is not crucial to the outcome of the method, as long as it reflects a certainty of similarity, (e.g. it take values close to 1), thus ensuring that the identified subset $\mathcal{V}_C$ is indeed a cluster core.

B. *Augmentation-Step*

What is needed now, is a systematic way of adding (previously eliminated) nodes to the identified subset of nodes, based on some selection criterion, with the goal of "forming" the rest of the cluster around its identified core $\mathcal{V}_C$. To this end let us define a sequence of node subsets of monotone increasing cardinality:

$$\mathcal{V}_a(k) = \mathcal{V}_a(k-1) \cup i_{max}(k), \quad \mathcal{V}_a(0) = \mathcal{V}_C \tag{6}$$

where

$$i_{max}(k) = \underset{i \in \mathcal{V} \setminus \mathcal{V}_a(k-1)}{\arg\max} R(i, \mathcal{V}_a(k-1)). \tag{7}$$

In other word, in each augmentation round, of all the candidate nodes that are not yet added, the one with the highest relevance to subset $\mathcal{V}_a(k-1)$, i.e. the node that determines the separation measure $S(\mathcal{V}_a(k-1))$, is selected for addition. Due to the starting point of the augmentation procedure, as well as the selection criterion of the node which is added in each round, we anticipate that the first nodes to be added in the core will be the remaining nodes of the cluster that will frame the initial core, followed by the rest of the nodes of the graph, which are irrelevant to the cluster at hand. Thus, what is needed to define in order to conclude our method, is a suitable decision rule that will help us to determine whether the subgraph which is formed during the augmentation procedure represents the complete class at hand (all the objects of the class have been included), in which case the augmentation procedure should be stopped, or not. More specifically our goal is the determination of the particular augmentation round, after which, the formed subgraph can no longer be considered as one cluster, i.e. not all the objects it represents belong to the same class. In order to detect this change, we propose the use of the following objective function:

$$C(k) = \frac{1}{k} \sum_{j=0}^{k-1} Q(\mathcal{V}_a(j)) - Q(\mathcal{V}_a(k)). \tag{8}$$

Since, as it is evident from (8), the first term of the proposed scheme constitutes the running average of the measure's sequence $\{Q(\mathcal{V}_a(j)),\ j = 0, 1, \cdots, k-1\}$, if the clusters comprising the given data set are well separated, then adding an "irrelevant" node to an already formed cluster (during the previous rounds), will introduce a steep drop in the $Q(\mathcal{V}_a(k))$ values, under the assumption that the newly added node will have much lower relevance to the nodes of the already formed subgraph. Hence, the new $Q(\mathcal{V}_a(k))$ value will be much lower compared to the previous ones.

Consequently, the detection of this particular round in the case of well-separated clusters is a fairly easy task. In real data set however, as this is not always the case, the sequence of $C(k)$ could attain many close$-$valued local maxima, leading to misclassification errors. Therefore, we propose the use of a more robust characteristic function, which constitutes a weighted version of $C(k)$ defined in (8), as follows:

$$C_W(k) = Q(\mathcal{V}_a(k))C(k). \tag{9}$$

Having defined a characteristic function with the desired characteristic, we can obtain the solution of the clustering problem, i.e., the identification of the cluster at hand, $\mathcal{V}_a(k^*)$, by solving the following maximization problem

$$k^* = \arg\max_k C_W(k). \tag{10}$$

After a cluster has been identified, then the subgraph induced by its nodes is eliminated from the initial graph, and the procedure is repeated having the newly formed graph as a starting point.

A formal outline of the proposed algorithm, in the form of pseudocode follows.

```
procedure CLUSTERIT($\mathcal{G}$)
Input graph $\mathcal{G} = (\mathcal{V}, \mathcal{E},\ W)$ and threshold value $\alpha$
$n = 0$
    while $|\mathcal{V}| > 2$ do
     $k = 1$
     $\mathcal{V}_e(0) = \mathcal{V}$
        while $Q(\mathcal{V}_e(k-1)) < \alpha$ do▷ Elimination Rounds
         $i_{\min} = \arg\min_{i \in \mathcal{V}_e(k-1)} R(i, \mathcal{V}_e(k-1))$
         $\mathcal{V}_e(k) = \mathcal{V}_e(k-1) \setminus i_{min}$
         $k = k + 1$
        end while
        $\mathcal{V}_C = \mathcal{V}_e(k-1)$        ▷ Isolation of Cluster's Core
        $k = 1$
        $\mathcal{V} = \mathcal{V} \backslash \mathcal{V}_C$                ▷ Remove the Core from $\mathcal{G}$
        $\mathcal{V}_a(0) = \mathcal{V}_C$
        while $|\mathcal{V}| > 0$ do               ▷ Augmentation Rounds
            $i_{max}(k) = \underset{i \in \mathcal{V} \backslash \mathcal{V}_a(k-1)}{\arg\max} R(i, \mathcal{V}_a(k-1))$
            $\mathcal{V}_a(k) = \mathcal{V}_a(k-1) \cup i_{max}(k)$
            $C_W(k)$                       ▷ Evaluation of Cost Function
            $\mathcal{V} = \mathcal{V} \backslash i_{\max}$
            $k = k + 1$
        end while
        $k^* = \arg\max_k C_W(k)$
        $n = n + 1$
        $\widehat{\mathcal{C}}_n = \mathcal{V}_a(k^*)$            ▷ The $n$-th Identified Cluster
        $\mathcal{V} = \mathcal{V}_a(k) \backslash \mathcal{V}_a(k^*)$
    end while
end procedure
```

Finally, concerning the complexity of the proposed method, as can be easily deduced by the above presented formal outline, the complexity of each iteration is in the order of $|\mathcal{V}|^2$, where $|\mathcal{V}|$ is the number of nodes of graph $\mathcal{G}$. Hence, assuming that k iterations are needed for the termination of the algorithm, i.e. k clusters are identified, then the overall complexity is in the order of $k|\mathcal{V}|^2$. Considering now the fact that in most meaningful applications $k \ll |\mathcal{V}|$, we maintain than the complexity of the proposed method is $O(|\mathcal{V}|^2)$.

IV. Simulation Results

In this section we evaluate the performance of the proposed clustering method, as well as its rivals, by applying it in both artificial as well as real data sets. Before proceeding with the presentation of our simulation results let us first briefly present the figures of merit we are going to adopt in order to compare partitions resulting from the application of the methods under comparison.

To this end let $\mathcal{X} = \{X_1, X_2, \cdots, X_{n_\mathcal{X}}\}$, $\mathcal{Y} = \{Y_1, Y_2, \cdots, Y_{n_\mathcal{Y}}\}$ be two partitions of the data set, with $n_\mathcal{X}$, $n_\mathcal{Y}$ denoting the corresponding number of clusters respectively.

As our first figure of merit we adopt the *Rand index* [2] that belongs in *pair counting* category of measures, and is defined as follows:

$$\mathcal{R}(\mathcal{X},\ \mathcal{Y}) = \frac{1}{1 + \frac{\alpha_{10} + \alpha_{01}}{\alpha_{11} + \alpha_{00}}}, \tag{11}$$

where α_{11} indicates the number of pairs of nodes which are in the same community in both partitions, $\alpha_{10}(\alpha_{01})$ the number of pairs of nodes which are in the same community in partition $\mathcal{X}$ $(\mathcal{Y})$ and in different communities in $\mathcal{Y}$ $(\mathcal{X})$ and finally, α_{00} the number of pairs of nodes which are in different communities in both partitions. As it is evident from its definition, *Rand index* takes values in the interval $[0,\ 1]$ with its maximum value indicating identical partitions.

As a second figure of merit we are going to use the *Normalized Mutual Information* [2] which has its roots on information theory and is defined as follows:

$$\mathcal{I}_{norm}(\mathcal{X},\ \mathcal{Y}) = \frac{2\mathcal{I}(\mathcal{X},\ \mathcal{Y})}{\mathcal{H}(\mathcal{X}) + \mathcal{H}(\mathcal{Y})}, \tag{12}$$

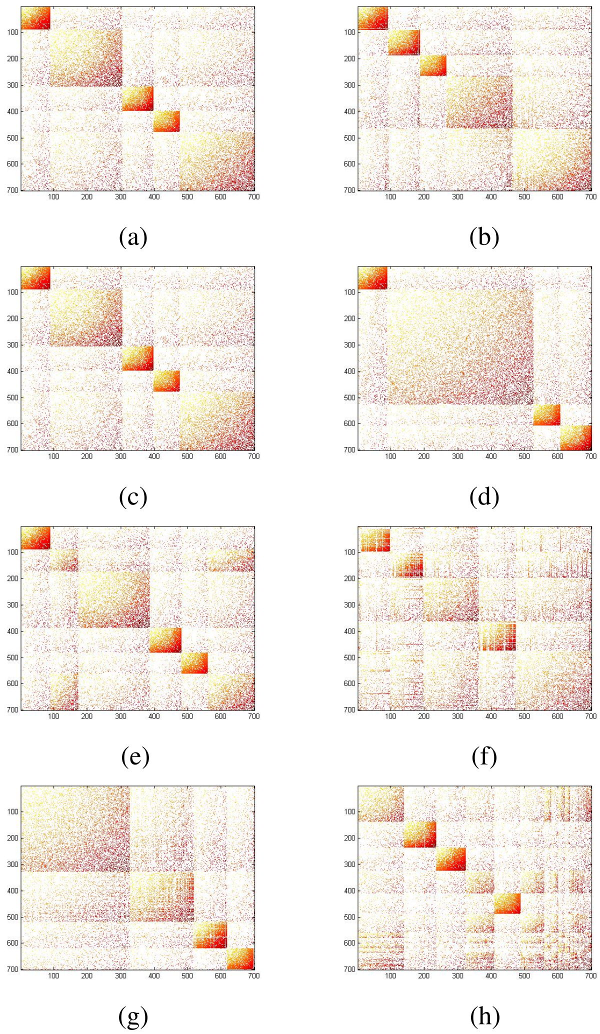
where $\mathcal{I}_{norm}(\mathcal{X},\ \mathcal{Y})$ denotes the *mutual information*, with partitions $\mathcal{X}$, $\mathcal{Y}$ considered as random variables, and $\mathcal{H}(\mathcal{X})$, $\mathcal{H}(\mathcal{Y})$ the *Shannon entropy* of $\mathcal{X}$ and $\mathcal{Y}$ respectively. As it is clear from its definition, *Normalized Mutual Information* takes its maximum value if the partitions under comparison are identical, while it takes its minimum value if the partitions are statistically indepented.

A. Experiment I

In this experiment we apply the proposed technique in synthetic data and compare its performance against well known clustering techniques, such as NCut, RAssoc, the single, complete and average linkage clustering techniques, and MCL. For the construction of a data set we used the software presented in [15] which constitutes the state of the art of graphs generator. The data set consisted of 500 graphs each having a cardinality of node set equal to 700. The size of the contained clusters varied in the range $[50,\ 250]$, average degree was 100 and both mixing parameters were set to 0.4. A sample of a similarity (ideally clustered) matrix as well as the results obtained from the application of the proposed method and its rivals with the best performance are shown in Fig. 1. Although Ncut results in an ideal clustering when the correct number of clusters is provided, as it is clear from Fig.1, its performance is vitally degraded if the given number of clusters is lower (Fig.1(d)) or higher (Fig.1(e)), even by as much as one. The same remarks can be stated concerning the sensitivity of MCL with respect to inflation parameter r. More specifically, while the clustering obtained for $r = 2.2$ coincides with the ideal one, a small perturbation in this value leads to greatly different results, as shown in Fig.1 (g),(h). This is more evident in Fig. 2 where the histograms of the adopted figures of merit are shown. Specifically, the histograms of *Rand index* (a) and *Normalized Mutual Information* (b) obtained from the application of the proposed (red), Ncut with the correct number of clusters given (green), MCL with $r = 2.2$ (magenta), and the partition with the best indices of the hierarchy (blue) technique respectively are shown. Note how the performance of the spectral based technique is degraded even for a small error in the given number of existing clusters, which is the rule in real applications where the number of clusters is unknown (Fig. 2, (c),(d)). Note also the dependence of the MCL results on the value of the inflation parameter (Fig. 2, (e),(f)).

B. Experiment II

The goal of this experiment is to indicate that there is a resolution limit when clustering is based on global optimization criteria. To this end a data set consisted by 500 graphs was constructed. The basic difference from the data set used in the previous experiment was the strong diversity of the cluster sizes. Each graph is composed by a number of clusters whose sizes are coming from two populations. Specifically, the first population is in the range 100 to 300, while the second one is in the range 10 to 50. In addition the edge weights were selected from two random populations; one with high values for intra-cluster edges and one with low values for inter-cluster edges. An example with the results we have obtained from the application of the techniques under comparison is shown in Fig.3. Moreover, as we can see from Fig. 4, where the obtained histograms of the adopted figures of merit are shown, in this kind of setup, the proposed method clearly outperforms its rivals thus revealing its insensitivity



(a) (b)

(c) (d)

(e) (f)

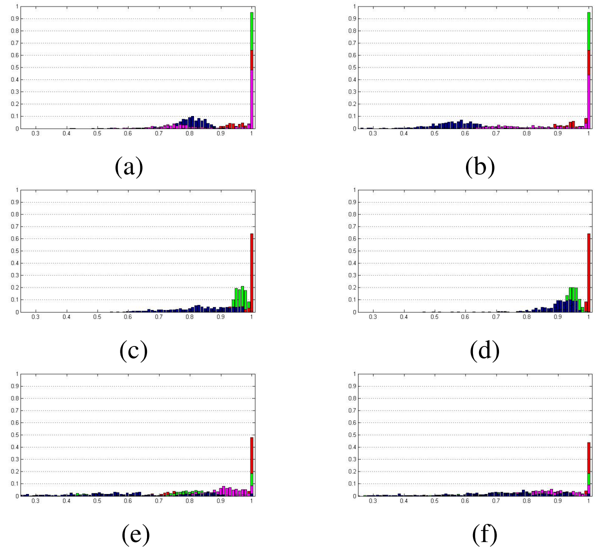
(g) (h)

Fig. 1: Ideal clusters (a) and clustering results obtained from the application of the proposed technique (b), NCut with $k = 5$ (the correct number of clusters) and MCL with $r = 2.2$ (c), NCut with $k = 4$ (d) and $k = 6$ (e), the "best" instance of *average linkage* (f), and MCL with $r = 2.0$ (g) and $r = 2.4$ (h), respectively.

regarding the number as well as the size distribution of the existing clusters. This is clearly not the case for Ncut, even though the correct number of the existing clusters was provided. MCL on the other hand, seems to be affected by the aforementioned factors to a lesser degree, helped by the proper selection of the inflation parameter, which as we have already mentioned, controls the granularity of the clustering results. In the given experiment, after a trial and error procedure (similar to Experiment I), a selection of $r = 6.3$ provided the best overall MCL performance. Finally, the very low performance of the hierarchical method is attributed to the high overlap degree of the edge weight populations.

C. Experiment III

In this last experiment, we apply the proposed technique in a real proteins clustering problem. To this end, we have selected from the UniProt database,[1] three well known protein groups; namely *Globines*, *E3ligase*, and *Histone H1/H5* families, with populations 33, 27 and 135, respectively and we have conducted the following experiment. We applied the

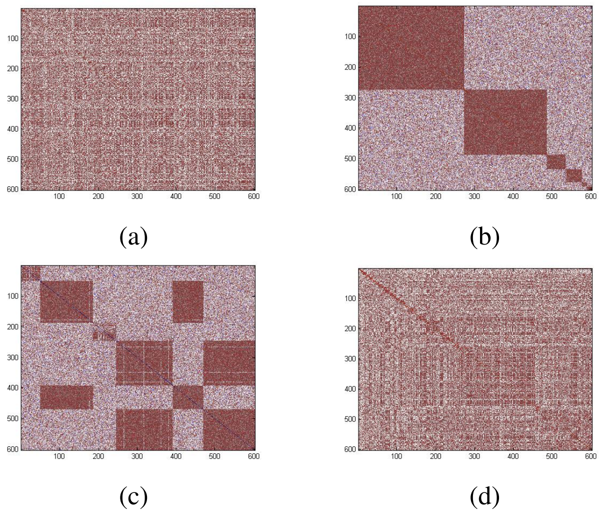


(a) (b)

(c) (d)

(e) (f)

Fig. 2: Histograms of *Rand index* (a) and *Normalized Mutual Information* (b) obtained from the application of the proposed (red), Ncut with the correct number of clusters given (green) and average linkage (blue) technique respectively. In subfigs (c) and (d) are the histograms of the same figures of merit, obtained from the application of the proposed (red) and Ncut technique with the given number of clusters being higher (green) and lower (blue) by 1, respectively. Finally, subfigs (e) and (f) display the histograms of the above mentioned metrics obtained from the application of MCL, with $r =$ 1.8 (blue), 2.0 (green), 2.2 (red), and 2.4 (magenta), respectively.

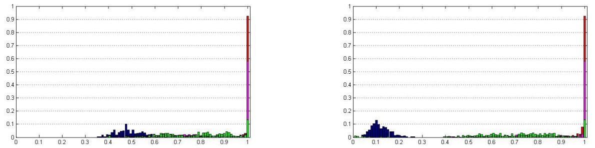


(a) (b)

(c) (d)

Fig. 3: A typical similarity matrix of the data set used in Experiment II (a) and the clustering results obtained from the application of the proposed technique and MCL with $r = 6.3$ (b), Ncut with the correct number of clusters given (c), and the best partition of the average linkage hierarchy (d) respectively.

[1] All similarity matrices were retrieved from the SIMAP database (*http://liferay.csb.univie.ac.at/portal/web/simap/*) using the provided *Submatrix export tool.*



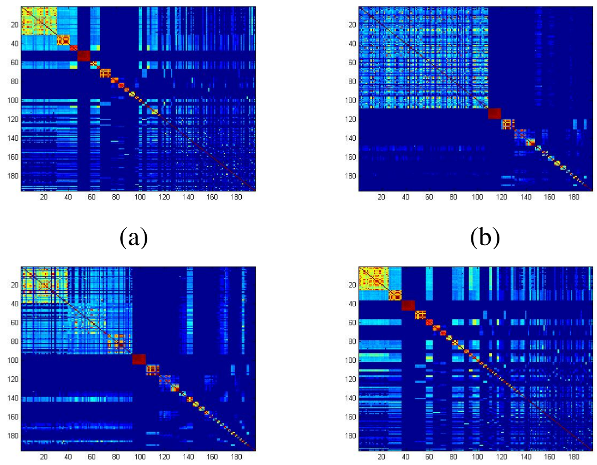
(a) (b)

Fig. 4: Histograms of *Rand index* (a) and *Normalized Mutual Information* (b) obtained from the application of the proposed (red), Ncut with the correct number of clusters given (green), MCL with $r = 6.3$ (magenta), and average linkage (blue) techniques, respectively.

proposed technique in each group of proteins separately and their clustering was obtained. More specifically, three clusters of size 6, 5 and 3 were identified from *Globines* group, four clusters of size 11, 9, 2 and 2 were identified from *E3ligase* and twelve clusters, with their sizes ranging between 2 to 30 were identified from the last protein *Histone H1/H5* group. Then, we applied the proposed method to the total similarity matrix resulting from an "all-against-all" comparison of the 195 proteins and the obtained clustering is shown in Fig. 5 (a). The proposed technique succeeds to identify exactly the same clusters identified from its application in the above mentioned groups separately, which is a clear indication of its robustness. The same experiment was also repeated using MCL for values of r ranging from 1.4 to 10. We observed that the number of the identified clusters (and the overall quality of the obtained clustering) increased with the increasing of r, up to a value (of $r \approx 6$), after which, further increasing r had no significant impact on the outcome. Clustering results obtained from MCL with $r = 1.8$ and $r = 10$ are depicted in Fig. 5 (b) and (c), respectively. In order to further improve these results, we proceeded thresholding the initial similarity matrix at different similarity levels, increasing this way the separation between clusters (essentially by removing bridge edges in the initial graph). We concluded that (for this particular example) threshold values between 0.4 and 0.6 presented the best balance between removing (unwanted) inter-cluster and (useful) intra-cluster edges. This procedure led to substantially improved clusterings, as well as a relative insensitivity to the value of r, giving the best results for $r \in [1.8, 5]$. As we can see from Fig. 5 (d), the "best" clustering of MCL is virtually identical to the result obtained by applying the proposed method in the initial graph (i.e. without adopting any thresholding scheme). Concluding, although the clustering example at hand can be considered as a small-scale one, the obtained results are very promising and indicative of the potential of the proposed technique.

V. Conclusion

In this paper a new iterative graph based clustering technique was proposed. Three desirable characteristics of the proposed technique were validated through a series of experiments both in artificial as well as real protein data sets. The advantages of the proposed technique against well known and widely used clustering techniques were demonstrated. Although a more exhaustive investigation is on the way, the experiments we have conducted so far, have led to very promising results regarding the problem of grouping together proteins based only on sequence information.



(a) (b) (c) (d)

Fig. 5: Clustering of the "all-against-all" similarity matrix, using the proposed technique (a), MCL with $r = 1.8$ (b), $r = 10$ (c), and MCL with $r = 1.8$ combined with proper thresholding (d).

Acknowledgement: The authors would like to thank the Laboratory of General Biology, Medical School, University of Patras, for their support in providing the experimental data set and for offering their expertise on several biological issues. This work was financed by the University of Patras, "Karatheodori" research program, entitled "*The relocation problem of seismic event hypocenter parameters*".

References

[1] J. A. C. A. Paccanaro and M. A. S. Saqi, "Spectral clustering of protein sequences," *Nucleic Acids Res.*, vol. 34(5), pp. 1571–1580, 2006.

[2] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, pp. 75–174, 2010.

[3] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, p. 026113, 2004.

[4] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," *in Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, 1967.

[5] B. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," *The Bell System Technical J.*, vol. 49, pp. 291–307, 1970.

[6] M. S. P. Chan and J. Zien, "Spectral k-way ratio cut partitioning," *IEEE Trans. CAD-Integrated Circuits and Systems*, vol. 13, pp. 1088–1096, 1994.

[7] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, pp. 888–905, 2000.

[8] Y. G. I. S. Dhillon and B. Kulis, "Weighted graph cuts without eigenvectors:a multilevel approach," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29(11), pp. 1944–1957, 2007.

[9] U. von Luxburg, "A tutorial on spectral clustering." *Technical Report 149, Max Planck Institute for Biological Cybernetics*, 2006.

[10] S. L. B. S. Everitt and M. Leese, *Cluster Analysis*. Arnold, London, 2001.

[11] W. H. E. Day and H. Edelsbrunner, "Investigation of proportional link linkage clustering methods," *Journal of Classification*, vol. 2, pp. 239–254, 1985.

[12] T. W. et al, "Partitioning biological data with transitivity clustering," *Nat. Methods*, vol. 7, p. 419420, 2010.

[13] V. M. et al, "High-quality sequence clustering guided by network topology and multiple alignment likelihood," *Bioinformatics*, vol. 28, pp. 1078–1085, 2012.

[14] S. van Dongen, "Graph clustering by flow simulation," *Ph.D Thesis*, 2000.

[15] A. Lancichinetti and S. Fortunato, "Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities," *Phys. Rev. E*, vol. 80(1), pp. 116–118, 2009.